

Stitching: Pen Gestures that Span Multiple Displays

Ken Hinckley¹, Gonzalo Ramos^{1,2}, Francois Guimbretiere^{1,3}, Patrick Baudisch¹, and Marc Smith¹

¹Microsoft Research
One Microsoft Way
Redmond, WA 98052
+1 425 703 9065

{kenh, baudisch, masmith}
@microsoft.com

²University of Toronto
40 St. George St.
Toronto, ON M5S 2E4
+1 (416) 946-8491

bonzo@dgp.toronto.edu

³University of Maryland
3267 AV Williams Bldg
College Park, MD 20742
+1 (301) 405 7952

francois@cs.umd.edu

ABSTRACT

Stitching is a new interaction technique that allows users to combine pen-operated mobile devices with wireless networking by using pen gestures that span multiple displays. To stitch, a user starts moving the pen on one screen, crosses over the bezel, and finishes the stroke on the screen of a nearby device. Properties of each portion of the pen stroke are observed by the participating devices, synchronized via wireless network communication, and recognized as a unitary act performed by one user, thus binding together the devices. We identify the general requirements of stitching and describe a prototype photo sharing application that uses stitching to allow users to copy images from one tablet to another that is nearby, expand an image across multiple screens, establish a persistent shared workspace, or use one tablet to present images that a user selects from another tablet. We also discuss design issues that arise from proxemics, that is, the sociological implications of users collaborating in close quarters.

Categories and Subject Descriptors

H.5.2 [Information Interfaces and Presentation]: Input Devices and Strategies, Interaction Styles

General Terms

Human Factors, Design

Keywords

pen computing, mobile devices, co-located collaboration, proxemics, synchronous gestures, spontaneous device sharing

1. INTRODUCTION

With wirelessly networked mobile devices rapidly becoming ubiquitous, it is increasingly important to find effective methods for dynamically binding together mobile devices in support of sharing information, collaborating and communicating with other persons, or performing complex tasks involving large documents and multiple sources of information. The *spontaneous device sharing problem* poses the following question: how can a user dynamically forge a purposeful connection between two or more

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

AVI '04, May 25-28, 2004, Gallipoli (LE), Italy
© 2004 ACM 1-58113-867-9/04/0500..\$5.00

target devices that do not *a priori* know one another's network address? This is not only a significant research problem for system implementation, interaction design, and social and behavioral observation, but also is becoming a problem with a pressing need for practical solutions.

Researchers have recently discovered a new class of techniques that use synchronous user actions (known as *synchronous gestures*) to address the spontaneous device sharing problem without forcing the user to manually enter network addresses. Examples of these techniques include holding two devices together and shaking them [13], bumping a pair of devices together [12], or simultaneously pressing a button on each device [19]. However, the design space of synchronous gestures is still not well understood and needs further exploration.

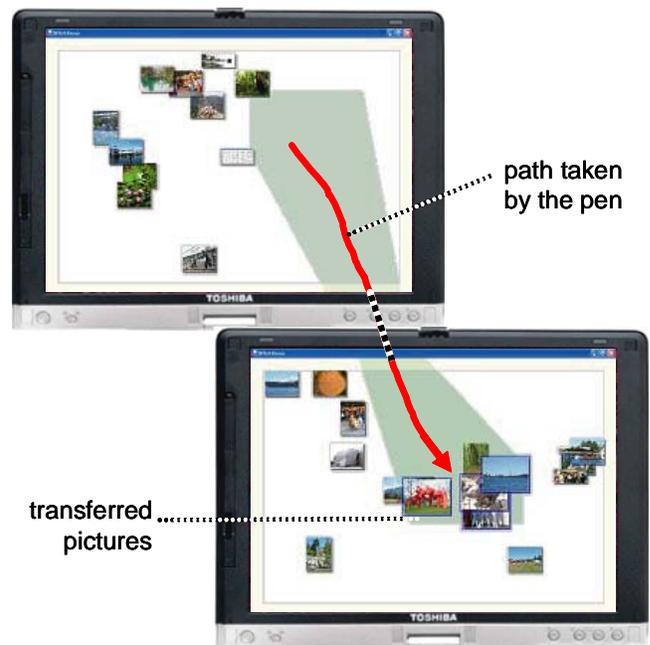


Figure 1. Here, a user gives some photos to another user by stitching from the top tablet to the bottom tablet.

Here, we contribute *stitching*, a new type of synchronous gesture that uses commonplace pen input capabilities to establish and manage serendipitous connections between pen-operated mobile devices. A stitching gesture is a pen gesture that spans multiple displays, consisting of a continuous pen motion that starts on one device, skips over the bezel of the screen, and ends on the screen of another device (Fig. 1). We show how stitching serves as the basis for a flexible connection architecture that can be

implemented on a variety of pen-operated devices. To explore the pragmatics of stitching, we implement a prototype photo sharing application for the Tablet PC that supports operations such as copying images from one tablet to another that is nearby, establishing a persistent shared workspace for collaboration, expanding an image across multiple screens, or using one tablet to display a slideshow of images that a user selects from another tablet. Preliminary usability testing suggests that users can easily make pen gestures that span displays, and find it compelling to have a straightforward means to perform cross-device operations.

In addition to the new technique of stitching, more generally we contribute six requirements for the spontaneous device sharing problem, phrased as design questions:

- **Connection:** How is a connection established?
- **Command:** What type of connection is required?
- **Operands:** What information is shared?
- **Geometry:** What is the spatial relationship between devices?
- **Coexistence:** How do connection gestures coexist with traditional interactions or naturally occurring user behaviors?
- **Proxemics:** How do users share physical space?

The literature presently lacks such a discussion of general requirements. In this sense our research represents a framework that considers cross-device operations in general, including the system, interaction, and social issues that arise.

2. RELATED WORK

Our work is the first to address all six requirements listed above. Several previous systems foster collaboration between ubiquitous devices, but require special hardware such as overhead cameras, visual identification tags, or radio-frequency identification (RFID) tags on each device [15,20,27,28]. Some systems have focused on new ways to use devices together, but may require manual configuration and calibration, including manual entry of network addresses [17] and the geometry of displays [14,26].

Some existing standards are relevant to the spontaneous device sharing problem. Bluetooth (www.bluetooth.org) supports device discovery, but offers no way to distinguish a target device from other devices within radio range: the user must search through a potentially long list of devices to find a symbolic name. Infrared Data Association (IrDA, www.irda.org) allows data transfer between two devices. However, IrDA requires that devices remain still during communication, and provides no flexibility in the relative position and orientation of the devices.

2.1 Synchronous Gestures

Synchronous gestures are distributed patterns of user activity that occur together in time, or in a specific sequence in time [12]. The literature currently describes three examples of synchronous gestures: bumping [12], Smart-Its Friends [13] and SyncTap [19]. Bumping devices together can be sensed and used to create a shared display that spans two or more devices [12]. Smart-Its Friends and SyncTap recognize when a gesture occurs on two devices at the same time. Smart-Its Friends infers a connection when two devices are held together and shaken. If a “friend” device previously identified in this manner comes within range of another device’s low-power radio, the device beeps, thus creating awareness of other nearby users. No other features are supported.

With SyncTap, a user can simultaneously press “Sync” buttons on two separate devices to connect them. SyncTap’s main strength is that it offers a lowest common denominator solution, since nearly all mobile devices could include a suitable Sync button. However, actually doing anything with the connection requires additional steps. For example, to establish cursor migration between a laptop and a desktop PC, the user must first perform a SyncTap gesture, and then switch to the mouse on the desktop computer to indicate the edge of the screen that should be linked to the other device. Stitching can support similar types of operations in a single fluid pen-based command phrase [4], without multiple steps that require switching devices. SyncTap also discusses the need to avoid *collisions* of multiple overlapping Sync actions; if a collision occurs, the user must repeat the gesture. This is a less critical issue for stitching since stitches can only collide if they exhibit the same timing *and* the same geometrical properties.

2.2 Pick and Drop

Pick and Drop [18] allows users to pick (copy) an item from one screen and drop (paste) it onto the screen of another nearby device. Stitching makes significant contributions to the literature beyond Pick and Drop, primarily by recognizing the requirements for a versatile interaction paradigm for combining multiple mobile devices. Our research identifies general design requirements of the spontaneous device sharing problem and illustrates how distributed pen input represents a rich design space that provides some elegant solutions addressing these requirements. Stitching provides users multiple ways to copy or move information between devices, link the displays of devices, and to otherwise use devices together. Stitching uses the geometrical information from the pen to automatically determine the spatial relationship between a pair of devices. This automatic calibration technique has not been previously implemented or reported in the literature. Our research is also the first to recognize that proxemics (how users share physical space) may yield critical design insights for the spontaneous device sharing problem.

Stitching also differs from Pick and Drop in its technical requirements. Pick and Drop requires a pen with an embedded unique ID, whereas stitching uses only a series of time-stamped pen events to infer a connection. This is a subtle technical difference but has tremendous practical implications: to our knowledge, there are *no* mobile devices available that support the pen ID feature required for Pick and Drop, whereas Stitching gestures can be supported between any pen-based mobile devices with compatible pen technologies, as summarized in Fig. 2. This technical difference may become less important if a pen ID capability becomes commonplace on future mobile devices.

	PDA (PocketPC, Palm)	Tablet (Tablet PC)
Stitch from PDA to...	<i>Yes</i> PDA/PDA stitches can use any stylus	<i>Yes</i> Tablet/PDA stitches must use the tablet’s stylus
Stitch from Tablet to...	<i>Yes</i> Tablet/PDA stitches must use the tablet’s stylus	<i>Sometimes</i> Tablet/tablet stitching requires compatible pens
Pen Technology	<i>resistive touchscreen</i> senses <i>any</i> solid object in contact with screen	<i>magneto-inductive</i> active digitizer senses only special pen

Figure 2. Summary of pen technologies used by Tablets and PDA’s, and the implications for stitching.

2.3 Proxemics and Social Issues

Proxemics is the study of how people use the invisible bubble of space that surrounds an individual [2,11,25]. Sociologist E. T. Hall [11] distinguishes two distances within arm's reach, intimate and personal, with social and public distances beyond that. No single distance between persons is "best" in all situations. Factors such as how well the persons know one another, the gender of each person, and what the persons are trying to accomplish can all influence selection of a social distance that feels comfortable to both persons [2,24]. It is often assumed that users prefer face-to-face interaction, but proxemics research suggests that people's choice of relative body orientation influences communication patterns and changes with the task at hand [7,11,21,24].

Although co-located collaboration involves users working in close physical proximity, this literature rarely mentions proxemics. Scott proposes territory-based techniques for tabletop interaction [22,23]. Rodden et al. [21] discuss social issues surrounding face-to-face consultations, but do not reference the sociological literature. Thus, we believe our discussion of proxemics is a contribution to co-located collaboration in general, as well as to the more specific problem of spontaneous device sharing.

3. THE REQUIREMENTS OF STITCHING

Stitching represents an explicit user command to combine the resources of multiple computers. We wanted stitching to provide a flexible and potentially extensible facility that would support a number of different ways of combining devices, rather than supporting only a single operation or a very limited set of options, as in previous systems (e.g. [13,18,28]). To achieve this, stitching addresses the following central design questions:

- **Connection:** How is a connection established? A user must name the devices that are involved in a multi-machine operation, and the system needs to provide feedback to the user(s) of those devices that a connection has been established.
- **Command:** What type of connection is required? The user needs to be able to choose among several possible ways to combine the devices. Does the user want to copy a file from one device to another? Establish a persistent shared workspace for collaboration? Expand an image across multiple screens? These all represent *multi-device commands* that transcend the barriers between devices.
- **Operands:** What information is shared? Multi-device commands may require operands, such as which file to copy to another computer. Users need mechanisms to select one or more objects as part of a stitching gesture.
- **Geometry:** What is the spatial relationship between the devices? Several previous systems support features, such as cursor migration or combining the screens of two devices, that require knowledge of where one display is relative to another [14,20]. Stitching uses the information provided by the pen to infer the spatial relationship between devices. This also allows us to provide graphical feedback for multi-device operations that appears to span devices, as seen in Fig. 1.
- **Coexistence:** How do connection gestures coexist with traditional interactions or naturally occurring user behaviors? In the case of stitching, connection gestures must coexist with existing uses for the pen including widget interactions,

inking, character entry, and naturally occurring human-human communicative gestures (such as waving the pen near the device while discussing the contents of the screen).

- **Proxemics:** How do users share physical space? Proxemics suggests that the arrangement of spaces can influence communication; as Hall writes, "what is desirable is flexibility... so that there is a variety of spaces, and people can be involved or not, as the occasion and mood demand" ([11], p 110). Interaction techniques that form impromptu associations between mobile devices should likewise support the range from users who know each other well and want to work closely together, to users who are strangers and may want to exchange files while maintaining social distance.

4. THE MECHANICS OF STITCHING

The above design questions suggest that stitching represents a new class of interaction techniques that could be implemented in a variety of ways. We now discuss the general concept of stitching in reference to a proof-of-concept photo sharing application, called *StitchMaster*, which we implemented for the Tablet PC. With digital photography becoming widespread, sharing photos with others is a task of interest to many persons. Also, many of the semantics that we wanted to explore with stitching, such as expanding an image across multiple screens or copying objects from one screen to another, represent useful and compelling operations for digital photographs. To begin, each user launches *StitchMaster* on his own tablet, which displays that user's photo collection as a set of thumbnail images.

4.1 Establishing a Connection

With stitching, a user names the devices to connect simply by moving the pen across them. Since there is a natural order implied by the gesture, stitching also establishes which machine is the sender of information, and which machine is the receiver. Some connection techniques are inherently bidirectional [13,19,28] and do not naturally provide this information.

In order to recognize a pen gesture that spans displays, a set of co-located mobile devices need to synchronize pen events (consisting of absolute pen (x,y) location on the screen, whether or not the stylus is in range of the screen, and a timestamp) with one another. There are a variety of well-understood technical means for devices to discover one another, so our research focuses on the *user interaction required to form a purposeful connection between two (or possibly more) specific target devices*, and not the system details of how background communication occurs between a set of co-located candidate devices. For example, *SyncTap* [19] simply multicasts button press timing information to all other devices on the same subnet. We could alternatively use a discovery mechanism such as that included in Bluetooth, or the Intentional Naming System [1]. We are also investigating the use of wireless signal strengths as a way to restrict synchronization communications to other nearby devices [3,12].

In our current prototype system, each participating device sends its pen events to a *stitching server*, which typically is hosted on one of the mobile devices (it may alternatively be hosted in the environment to offload computation from the mobile devices). The stitching server synchronizes time between the devices [9,12] and looks for matching pen traces; when a match is found, the server sends a stitching event that informs the two devices of each

other’s network address. In our current prototype each participating device must know the network address of the server, but this is the one and only address needed to bootstrap the system. This is sufficient for prototyping and user testing, but in order to make a deployable version of our system, we will need to implement a discovery mechanism.

4.1.1 Stitching Recognition

The stitching server recognizes a stitch by looking at the patterns of pen activity from each pair of participating devices. We define an *envelope* as the time interval during which the pen is in range of the screen and is moving at a speed above a predetermined threshold. The stitching server then looks for two consecutive envelopes from a pair of devices that match a specific pattern, as illustrated conceptually in Fig. 3.

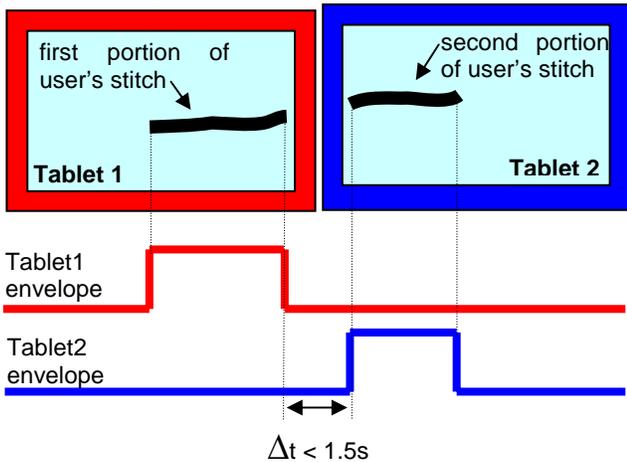


Figure 3. Conceptual diagram of stitching recognition.

The specific criteria used to recognize this pattern are:

- The first envelope must end near the first screen’s border and last longer than a timeout (250 milliseconds).
- The second envelope must start near the second screen’s border, and last longer than a second timeout (100ms).
- The second envelope must start after the first envelope, but it may occur no longer than 1.5s after the first envelope. This time interval is long enough to support stitching between tablets within arm’s reach (a maximum of about 75 cm).
- The direction of the pen as it exits one screen and enters another must match within $\pm 20^\circ$.

We found these criteria suffice to recognize intentionally executed stitching gestures, but just as importantly, false positives were not a problem: incidental pen motions from two users concurrently using pens on their mobile devices rarely satisfy these criteria. The main limitation of our stitching recognition algorithm is that it is difficult to perform a stitch that starts close to an edge of the screen; the system does not receive sufficient samples of the changing pen location, before it leaves proximity, to be certain of whether the user is lifting the pen from the screen, or whether the user is sliding the pen towards another device to perform a stitching gesture.

It may be possible to design variants of stitching that do not require the system to “infer” when a gesture is a stitch or not, by having users explicitly signal stitching gestures. For example, a

user could select a menu command such as *Stitch to Another Device...* before starting a stitch, and the user of the second display would select a corresponding menu command such as *Accept Stitch from Other User*. However, this approach seems to require introducing modes to the interaction. We focus on a recognition-based approach since it seems to work well, and makes the operation of stitching as simple as possible.

4.1.2 Physical Distance (Range) of Stitching

As we discuss further in our usability study (section 5), users do not want to be required to place their devices into persistent contact in order to form a connection; transient entry into personal space is acceptable, but touching is taboo in non-contact cultures [11]. For this reason, we designed stitching such that users can place their devices together if desired for close collaborative work, but stitching *does not* require that devices be in direct contact (immediately adjacent to one another) in order for the stitching gesture to be recognized. As noted above, we designed the software with time-outs so that a user can stitch between any two devices within arm’s reach. Most figures in this paper show two devices in contact, but this is only for pragmatic reasons, to allow us to show as much detail as possible in the screen captures.

4.1.3 User Feedback for Stitching

A stitch is recognized as soon as the first 100 milliseconds of the second envelope have been observed by the stitching server; it does not wait for the user to finish the motion. Performing this eager recognition allows us to provide a user with feedback of the stitching gesture as soon as possible after the pen has entered the second screen.

Feedback for a successful stitch consists of a short chirp sound as soon as eager recognition takes place. If the stitching gesture includes any operands, then the system shows a semi-transparent blue shadow on the screen in the shape of the selected photos (Fig. 4a). Upon completion of the stitching gesture, the system may also provide additional feedback. For example, for a copy or move operation, StitchMaster shows an animated semitransparent cone that appears to whisk files from one machine to the other (Fig. 1). This provides clear feedback of where the files came from, and where they were copied to (Fig. 4b).

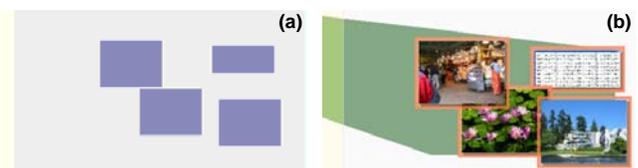


Figure 4. Feedback for remote copy. (a) Shadows appear on the other device, then (b) when the user drops the photos, a cone connects them to their origin on the other device.

4.2 Specifying Connection Type: Multi-Device Commands

We believe stitching is unique among proposed spontaneous device sharing techniques in that it includes a general command selection mechanism as part of the connection gesture. Multi-device commands currently supported by StitchMaster include copying or moving photos, establishing a persistent shared work space, expanding an image across multiple displays, or entering a

presentation mode known as the *gallery* (described below). StitchMaster presents these options in a marking menu [16], so it would be easy to add new multi-device commands to the system. There are several design choices for where the command selection can occur, depending on *when* the choice is made in the articulation of the stitch, and *where* the selection is made:

- *Local prefix menus*: Users choose a command (e.g. Copy) on their local screen, and then stitch to indicate the remote device that is involved.
- *Remote postfix menus*: Users stitch to another device, and then a menu appears on the remote device providing options for how to combine the devices.
- *Local postfix menus*: Users stitch to another device, return the pen to their own screen, and finally choose a command from a menu that appears on their own local screen.
- *Remote prefix menus*: This option would require the user to select a command on another screen, and then return to the local screen only to stitch back to the remote machine. From the user’s perspective this would be awkward; from the system’s perspective, it would be indistinguishable from a local prefix menu that originates on the other screen.

StitchMaster currently implements remote postfix menus, which allows us to limit the visibility of multi-device operations to situations where they are known to be applicable; we did not want to complicate the single-device user experience with options for multi-device operations. Remote menus appear at the end of a stitching gesture when the user holds the pen still for 0.5 seconds. To provide feedback that a menu is a *remote* menu, StitchMaster shows a transparent blue cone that connects the remote menu back to the display where the stitching gesture originated (Fig. 5). We are also experimenting with local postfix menus as an alternative (these are discussed further in our usability testing section).

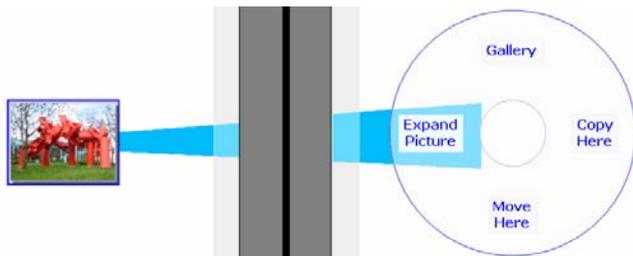


Figure 5. A remote menu shows a link between screens.

For some stitching gestures, StitchMaster assigns a default operation, eliminating the need to use the menus. For example, when stitching with a selected photo (that is, stitching using an operand as described in the next section), by default the selected photograph is *moved* to the other screen. We chose not to make *copy* the default since we found during pilot studies that users would repeatedly copy files back and forth while trying out stitching, quickly creating cluttered screens for themselves.

4.2.1 Example Multi-Device Command: The Gallery

The Gallery (Fig. 6) allows one user to give a presentation of selected photos to another user. To start the Gallery, the presenter selects an image to start with, stitches to the other screen, and chooses Gallery from the remote menu. The other tablet then displays a full-screen view of the selected image, while the

presenter’s tablet displays thumbnails of all of his photos. The presenter can click on any thumbnail to change the image that is displayed on the other tablet.



Figure 6. Gallery: The right tablet displays a full-screen view of an image that the presenter selects on the left tablet.

The Gallery changes the roles of the devices. Instead of two identical devices, we now have one tablet for interaction, while the other primarily serves as a display. If users separate the devices, but keep the Gallery running, the presenter’s tablet becomes a private view, while the other tablet represents a public view of selected information.

4.3 Specifying What to Share: Stitching with Operands

StitchMaster supports tapping on a single photo to select it, or drawing a lasso to select multiple photos. StitchMaster outlines the selected photos in orange and scales them to be slightly larger than the others (Fig. 7). Users can select a photo and then perform a stitching gesture to another device all in one gestural phrase [4,6,10]. The user makes a selection, and then lifts the pen slightly so that the pen is no longer in contact with the screen, but is still within tracking range of the Tablet PC screen. The user then stitches to the other display, and the selection is treated as the operand of the stitching gesture. Alternatively, the user may keep the pen in contact with the screen: the user makes a selection, briefly pauses (stops moving the pen), and then completes the selection to the other screen. The pause is used to segment the selection portion of the gesture from the stitch to the other screen.

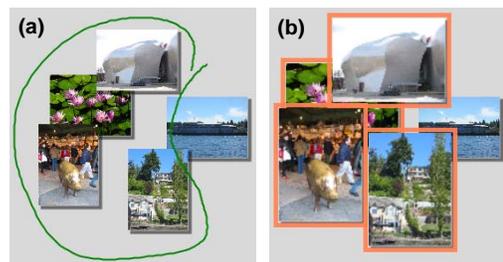


Figure 7. (a) Multiple selection using the lasso gesture. (b) Selected photos scale up and highlight in orange.

Note that we do not *require* stitching to follow selection in a single uninterrupted gestural phrase. A stitching gesture that starts over a selection also includes that object as an operand, but after 3 seconds, the selection *cools* and will no longer be treated as the operand for a stitching gesture. The highlights around selected photos turn blue once the selection has cooled. This approach prevents old, stale selections from mistakenly being interpreted as the operand to a stitching gesture.

4.4 The Spatial Relationship between Devices

To infer the spatial relationship between devices, stitching fits a line equation to the coordinates traversed by the pen on each screen. Of course, users do not move the pen in perfectly straight lines, but users do tend to move in arcs that can be locally approximated by a straight line.

When the stitching server detects a stitch from Device1 to Device2, it records a small window of samples as the pen leaves one screen and enters another, yielding p_0 (the exit point of the first pen trace), p_{10} (the entry point for the second pen trace), p_{11} (the point at which the stitch was recognized), and α_0 (the angle of motion at p_0); see Fig. 8. If the devices do not face the same way, then we rotate Device2's parameters to correct for this.

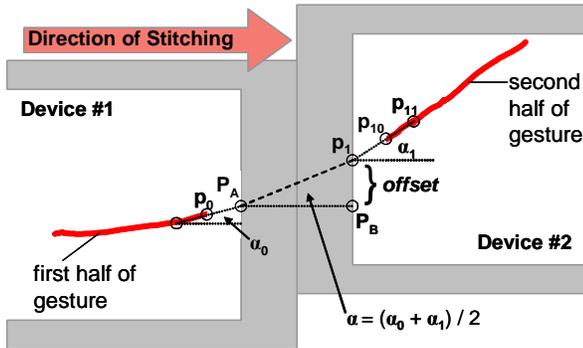


Figure 8. Fitting a line to the user's pen gesture.

Due to the sampling rate of the pen, the first and last pen locations reported by the tablet may fall up to 3-4 cm from the edge of the screen. We found that calculating the distance between the devices by using the time interval between the last and first observed samples was unreliable. For this reason, we ignore any empty space that may be present between the devices. This works extremely well in practice, since we find that users focus on the two screens and tend to ignore any empty space between them.

We estimate the intersection of the stitching gesture with the edge of each screen, yielding the points P_A and p_1 . P_A is the intersection of the screen edge of Device1 with the line that passes through p_0 at an angle α_0 ; p_1 is the intersection of the second screen edge with line that passes through p_{10} and p_{11} at angle α_1 . If the line between P_A and p_1 has angle α , the *offset* between the two screens is then $\tan(\alpha)$ times the bezel width. We estimate α as the average of α_0 and α_1 , which seems to work well, even if the user follows an arcing path while stitching. We then calculate P_B as the displacement of p_1 along the edge of Device2's screen by *offset* pixels. Using this approach, our system can transform points from one device's coordinate system to the other, thus allowing the presentation of graphics that appear to span the devices.

4.5 Coexistence of Stitching with Traditional Interactions

Stitching must allow users to establish connections between devices without interfering with existing uses for the pen. Input states supported by pens [5] include *tracking* (moving the pen in close proximity to the screen, causing the tracking symbol to move), *dragging* (moving the pen in contact with the screen, causing an action such as dragging an object or leaving an ink trail), and *out-of-range* (the pen is not in the physical tracking

range of the screen). Stitching can be implemented using the dragging state, or using the tracking state. StitchMaster implements options to use either style of stitching, or both can be supported simultaneously (this is the default on the Tablet PC).

4.5.1 Stitching in the Dragging State

Since traditional GUI interactions occur in the dragging state, performing stitching by dragging could conflict with them. For example, when stitching via dragging, the first device cannot be sure whether to interpret a pen stroke as a drag until the second device recognizes the completion of the stitching gesture. To circumvent this problem and allow stitching via dragging to coexist with other dragging operations, we use speculative execution: StitchMaster initially assumes all pen strokes are intended as drags. If the stitching server then reports a stitch, StitchMaster undoes the drag and instead treats the gesture as part of a stitch.

During preliminary user testing, we found that users can easily make a stroke while keeping the pen in contact with the screen. When stitching to another display, users quickly learn to "jump" the pen over the bezel, so in practice the bezel does not get in the way or cause significant problems for users.

4.5.2 Stitching in the Tracking State

Stitching from the pen's Tracking state represents a more advanced skill than dragging, as it requires moving the pen while keeping the tip within 2 cm of the screen surface to prevent it from entering the *out-of-range* state. However, by resting one's palm on the screen while holding the pen just above the screen surface, it is easy to make a continuous, quick, and fluid movement that is not interrupted by the physical "speed bump" of the screen bezel. Another advantage of stitching in the tracking state is that it avoids the need for a speculative execution scheme: stitching gestures occupy a separate layer that rests on top of GUI interactions.

The main drawback of implementing stitching in the tracking state is that currently available personal digital assistants (PDA's) do not support tracking, so stitching gestures on PDA's must use the dragging state. Another *potential* problem with stitching in the tracking state is that users may use the pen to gesture while talking about the contents of the screen with a colleague, possibly resulting in false-positive recognition a stitch. However, naturally occurring user pen motions of this sort do not seem to match the criteria that we use to recognize stitching gestures: in particular, they do not cross the bezel of two different screens. Thus, in practice, we have not observed this to be a problem.

4.6 Sharing Physical Space

With varying social and cultural conventions, individual preferences, and changing needs depending on the task, users need flexible ways to share physical space when combining devices. Hall distinguishes two distances within arm's reach, intimate and personal, with social and public distances beyond that [11]. StitchMaster includes support for configuring spaces at the intimate, personal, and social distances.

Intimate spaces. Intimate spaces support tight collaboration between friends or colleagues who may need to work together on a large document. For example, StitchMaster supports placing two

tablets together and then expanding an image to fill both screens. The displays act as tiles of the same virtual space. This style is also well suited for a single user wishing to extend his workspace with additional devices.

Personal spaces. Users can stitch together tablets that are separated by up to about 75 cm. This allows a space created by stitching to act as a whole, yet each user maintains his or her own personal space. For example, StitchMaster allows users to create a persistent shared workspace by making a “simple stitch” from one screen to another without any operands. A red frame then appears to surround the two screens to give users ongoing feedback that the two machines are connected. Either user has veto power over the connection and can close the workspace by choosing Disconnect from a menu.

Social spaces. Once users join a shared workspace, they can further separate their devices, yet still work together at distances beyond arm’s reach. For example, a user can employ *the transporter* to give photos to the other user, even if that user is no longer within arm’s reach. The user drags a photo to the edge of the screen, and dwells with the pen. After a brief pause, during which we display an animation of a collapsing blue square, the photo is transported to the other device. This pause separates transporting a photo from drag operations; the collapsing blue square gives feedback that the picture is about to be transported.

Public spaces. Stitching does not include any techniques designed for use at the public distance (which Hall defines as beyond 4 meters). Our goal with stitching is to provide effective means for sharing devices and information within co-located groups; the public distance stretches the limits of what we would consider the bounds of a single co-located group.

Orientation of spaces. Research suggests that communication patterns change when persons sit face-to-face, shoulder-to-shoulder, or corner-to-corner (at 90 degree angles) [11,21,24]. Face-to-face interaction is often assumed to be optimal in the collaboration literature, but Sommer notes that corner seating preserves closeness while avoiding excessive eye contact; furthermore, students studying together strongly prefer side-to-side seating [24]. To support spontaneous collaboration in these situations, we have recently extended stitching to support pen gestures that span tablets in any of these orientations.

5. USABILITY TESTING

We conducted a usability study of StitchMaster to identify usability issues and user concerns with stitching. Our primary goal was to verify if users could effectively use stitching gestures to perform multi-device operations.

Participants: 12 participants were recruited from the general public through Microsoft’s usability pool. Since stitching is intended to allow users to share information (or their mobile device itself) with other users, all subjects were run in pairs. None of the paired participants knew each other prior to the study. Collaborating pairs were of the following genders: 1 pair female-female, 3 pairs male-female, and 2 pairs male-male. Four of the participants had previously used a Tablet PC; 6 participants had previously used some other type of pen-based device.

Materials: We ran the study on Toshiba Portege 3500 Tablet PC’s with built-in 802.11 wireless networking. These devices measure

28.5 x 23cm with 25 x 18.7cm screens. Users employed the tablets in the slate (flat) mode.

Procedure: The participants sat shoulder-to-shoulder on the same side of a table; the experimenter sat at the opposite side of the table. Each participant was provided with a Tablet PC running StitchMaster. In a 2-3 minute practice session, the participants learned basic pen operations such as selecting images and dragging images on the screen. The experimenter then explained how to use features of StitchMaster, and allowed participants to try them out one-by-one, but did not show users what to do. Each session lasted approximately one hour. Testing was divided across two days, with 3 pairs of subjects run each day. Some bugs and usability issues were corrected between days, as noted below.

5.1 Results

The experimenter first asked participants to “connect the devices by making a pen stroke across the devices” but did not show participants how to do this. With this instruction, all 12 participants, on their first or second try, created the *persistent shared workspace* by stitching with no operands.

All but two users made their first attempt at stitching by moving the pen in contact with the screen. Participants expressed no clear preference for performing stitching in the tracking state versus the dragging state; both seemed to work well. All participants at some point during the study used stitching in both manners, and users often would mix styles within the same gesture: for example, a user would perform the first half of a stitch by dragging, but then jump the bezel and complete the stitch from the tracking state.

The experimenter next explained how to move files between devices by stitching with operands. Originally, StitchMaster required the user to place the pen in contact with the other screen at the end of the stitching gesture to drop a photo. On our first day of testing, we found this was problematic for users who mixed the dragging and tracking styles of stitching. Users repeatedly moved the pen *away* from the screen when they wanted to drop a photo, rather than touching the screen. We fixed this problem for the second day of testing by having the software drop the photo if the user lifted the pen at the end of the stitch. For example, on the second day of testing, one user commented that “it was nice to drag items to the other screen without having to touch it.”

While moving a photo to the other screen, participants sometimes would pause too long, causing the remote menu to appear. Increasing the time-out for the remote menu reduced the number of subsequent cases of accidental activation, but did not completely eliminate this problem.

Participants sometimes failed to stitch if they started too close to the edge of the screen. Participants wanted feedback of how far from the edge of the screen they had to start stitching in order to be successful. Adding 1-2cm margins would make this limitation visible, and prevent users from leaving photos at the extreme edges of the screen where this problem arises.

The only instances of false positive recognition of stitches that we observed occurred if users failed to successfully stitch, and then returned the pen to their screen to try again. Without realizing it, users often returned the pen to their original screen while remaining in the tracking state, and this was sometimes recognized as a stitch.

Overall, users were enthusiastic about the concept of stitching as embodied by StitchMaster. When asked if “I would use this software if it were available” the average response was 6.7 out of 7. However, one area of concern for many participants was security and privacy. For example, participants wanted to know if “Once connected, can a person take my other stuff?” or if there was a “lock-out for security and privacy.” Currently, there is not.

5.2 Proxemics of Device Sharing

Our usability testing led us to two additional “lessons learned” in relation to proxemics:

5.2.1 Do Not Require Contact

We began testing sessions by instructing users to “put your tablets together.” Although many users followed these directions, some users seemed hesitant to place their tablet in direct contact with that of the other user. In 3 of the 7 sessions, participants placed their tablets together, but asked “Do they have to be right next to one another?” When the experimenter replied that they did not, subjects moved them approximately 15 to 40 cm apart. This appears to represent a comfortable separation of devices for users who sit shoulder-to-shoulder. Clearly, stitching must support gestures between tablets that are not immediately adjacent to one another. This is why we designed stitching recognition to work between any two devices within arm’s reach (section 4.1.2).

However, this does not mean that intimate spaces, with the devices close to or in contact with one another, are not useful. Users who know each other and are trying to closely collaborate may need to make use of intimate spaces (as discussed in section 4.6). When asked if “Combining the screens of multiple Tablet PCs was a compelling capability,” the average response was 6.8 (agree) out of 7. Users commented that they liked “the ability to split the view, so there are no two faces trying to peek at only one screen” and that the “wide screen would be nice for collaboration, like for two people working on the same spreadsheet.” Thus, although participants in our study worked with a stranger, they envisioned contexts where close, joint work would be valuable.

5.2.2 Establish and Relax

Users want to *establish* a connection via stitching, but then *relax* the increasing social tension by quickly exiting the personal space of the other user. In our study, when one user reached over with the stylus, the other user would often lean back to make the intrusion into personal space less acute. Many subjects made short stitching gestures that only extended 3-5 cm onto the other user’s screen, and some users held the pen near the top, like a pointing stick, rather than holding it at its tip, like a writing instrument. Users may have adopted these behaviors in an effort to minimize intrusions into the other user’s personal space. Similarly, the transporter, which allows users to share files without reaching into the other user’s personal space, was popular with test users.

Although participants successfully used remote menus to choose how to combine devices, this perspective does offer an argument against the use of remote menus, which require the user to perform command selections while reaching onto another user’s display. To avoid this, we are experimenting with *local postfix menus*, which allow the user to stitch, return to his local screen, and then select the multi-device command to execute. Further

usability testing will be required to see if users prefer this approach.

6. DISCUSSION

6.1 Security and Privacy

Security was one area of concern for some test users. Since only nearby persons can connect to a device, stitching does offer some inherent security measures. Social rules are at play, and because of the physical nature of the gesture, users who violate these rules by reaching onto a user’s screen without permission are likely to be noticed. Test users often verbalized their intent to stitch to another user’s screen; for example, one user commented “here’s a care package for you” before moving files to the other user’s screen. Nonetheless, users in an untrustworthy environment may wish to “lock out” stitching gestures, accept stitches only from devices which have previously been granted permission, or require a password. SyncTap discusses encrypted communication for spontaneous device sharing [19]. The Speakeasy system carefully considers access control and privacy [8].

6.2 Multi-Device Stitching

We have recently extended our stitching system architecture to support formation of sets of up to 16 devices, but StitchMaster currently only supports formation of pairs of tablets. The stitching server adds a device to a connected set if the user stitches between a connected device and a new, disconnected device. We also plan to experiment with *long stitches* that traverse more than two devices, connecting them all in one continuous gesture.

6.3 Stitching on PDA’s

Currently available PDA’s cannot sense the tracking state, but we have implemented stitching for PDAs using the dragging state. Our software can successfully recognize PDA stitching gestures, but we have not yet ported our photo sharing application to this platform. However, including both Tablets and PDA’s in our system may allow interesting new applications. Recall that stitching between Tablets and PDA’s is possible as long as the user employs the Tablet’s pen to perform the gesture (Fig. 2).

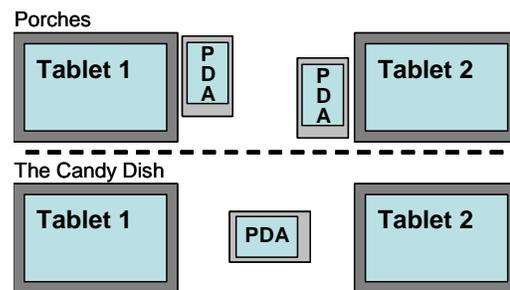


Figure 9. Changing the proxemics of file sharing by using tablet computers and PDA’s together.

For example, we have considered designs that use PDA’s to alter the proxemics of stitching. *Porches* (Fig. 9, top) use PDA’s as public areas for receiving visitors; each tablet has its own porch. To give a file to someone else, a user moves it onto the other user’s “porch” via stitching, or to offer a file for taking the user leaves it on his own porch. The other user can then take the file from a public porch into the more closely held main screen area. This reduces the need for each user to violate the personal space

of the other user. *The candy dish* (Fig. 9, bottom) places a single PDA in the no-man's-land between two other devices. Each user may then place files into the PDA via stitching, or take files that have been left there by the other user. Again, the users would not have to reach into each other's personal space to share files. Finally, *the shuttlecock* (not pictured) uses the PDA itself as a physical transport for data, in a manner analogous to the mediaBlocks system [29]. For example, a user could stitch to a PDA, pass the PDA to someone else, and the other user could then stitch to remove the data from the PDA.

6.4 Cooperative Stitching

While demonstrating stitching, we have noted that one user will sometimes draw the first part of the gesture on his screen, while a second user completes the second half of the gesture on another device. We call this *cooperative stitching*. Cooperative stitching allows users who are co-located, but beyond arm's reach of one another, to perform stitching gestures. In future work we would like to explore using cooperative stitching as a natural metaphor for supporting many-to-one connections. For example, to pass out a document at a meeting, a user could start a stitch, and then several other users could accept the document by completing the stitch on their own screens. Furthermore, since the pen never leaves the owner's device, the user does not have to worry about whether or not the pen is compatible with the other user's device.

7. CONCLUSION

We believe that the true untapped potential of the emerging wireless network lies in dynamic peer-to-peer coordination between proximal devices. Stitching provides an example of this perspective by offering users a versatile means to dynamically bind together pen-operated devices. We have provided some examples of multi-device commands, such as copying photos between devices, expanding an image across displays, creating a shared workspace, or using the gallery to project selected photos on another user's display. It is our hope that by identifying interaction requirements and usability issues for this new class of distributed pen interfaces, our work may spark further exploration of new applications, capabilities, and interaction techniques that foster communication, sharing, and collaboration between users of mobile devices, and empower users with new ways to combine the capabilities of multiple mobile devices.

8. REFERENCES

- [1] Adjie-Winoto, W., Schwartz, E., Balakrishnan, H., Lilley, J., *The design and implementation of an intentional naming system*. Proc. 17th ACM Symp. on Operating Systems Principles, 1999. **33**(5): p. 186-201.
- [2] Altman, I., *The Environment and Social Behavior*. 1975, Monterey, CA: Brooks/Cole Publishing Company.
- [3] Bahl, P., Padmanabhan, V., *RADAR: An In-Building RF-Based User Location and Tracking System*, IEEE Computer and Communications Societies: INFOCOM 2000, 775-84.
- [4] Buxton, W., *Chunking and Phrasing and the Design of Human-Computer Dialogues*, Proc. IFIP Information Processing '86: Amsterdam: North Holland, 475-480.
- [5] Buxton, W., *A three-state model of graphical input*, Proc. INTERACT'90: Amsterdam: Elsevier Science, 449-456.
- [6] Buxton, W., Fiume, E., Hill, R., Lee, A., Woo, C., *Continuous hand-gesture driven input*, Graphics Interface '83, 191-195.
- [7] Deasy, C.M., Lasswell, T.E., *Designing Places for People: A Handbook on Human Behavior for Architects, Designers, and Facility Managers*. 1985, New York: Watson-Guptill.
- [8] Edwards, W.K., Newman, M.W., Sedivy, J., Smith, T., Balfanz, D., Smetters, K., Wong, H.C., Izadi, S., *Using Speakeasy for Ad Hoc Peer-to-Peer Collaboration*, CSCW'02, 256-265.
- [9] Elson, J., Girod, L., Estrin, D., *Fine-Grained Network Time Synchronization using Reference Broadcasts*, SIGOPS Operating Systems Review, 2002. **36** (SI): 147-163.
- [10] Guimbretiere, F., Winograd, T., *FlowMenu: Combining Command, Text, and Data Entry*, UIST 2000, 213-216.
- [11] Hall, E.T., *The Hidden Dimension*. 1966, New York: Doubleday.
- [12] Hinckley, K., *Synchronous Gestures for Multiple Users and Computers*, UIST 2003, 149-158.
- [13] Holmquist, L., Mattern, F., Schiele, B., Alahuhta, P., Beigl, M., Gellersen, H., *Smart-Its Friends: A Technique for Users to Easily Establish Connections between Smart Artefacts*, Ubicomp, 2001: Springer-Verlag, 116-122.
- [14] Johanson, B., Hutchins, G., Winograd, T., Stone, M., *PointRight: experience with flexible input redirection in interactive workspaces*, UIST 2002, 227-234.
- [15] Kohtake, N., Rekimoto, J., Anzai, Y., *InfoStick: An Interaction Device for Inter-Appliance Computing*, Handheld and Ubiquitous Computing (HUC'99), 246.
- [16] Kurtenbach, G., Sellen, A., Buxton, W., *An empirical evaluation of some articulatory and cognitive aspects of marking menus*. Human Comput. Interaction, 1993. **8**(1).
- [17] Myers, B.A., *Using Hand-Held Devices and PCs Together*. Communications of the ACM, 2001. **44**(11): p. 34 - 41.
- [18] Rekimoto, J., *Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments*, UIST'97, 31-39.
- [19] Rekimoto, J., Ayatsuka, Y., Kohno, M., *SyncTap: An Interaction Technique for Mobile Networking*, Mobile HCI 2003, 2003: Springer, 104-115.
- [20] Rekimoto, J., Saitoh, M., *Augmented Surfaces: A Spatially Continuous Work Space for Hybrid Computing Environments*, CHI'99, 378-385.
- [21] Rodden, T., Rogers, Y., Halloran, J., Taylor, I., *Designing novel interactional workspaces to support face-to-face consultations*. CHI 2003, 57-64.
- [22] Scott, S.D., *Territory-Based Interaction Techniques for Tabletop Collaboration*, UIST'2003 Companion, 17-20.
- [23] Scott, S.D., Grant, K.D., Mandryk, R.L., *System Guidelines for Co-located, Collaborative Work on a Tabletop Display*, ECSCW'03, European Conference on Computer-Supported Cooperative Work, 159-178.
- [24] Sommer, R., *Further studies of small group ecology*. Sociometry, 1965. **28**: p. 337-348.
- [25] Sommer, R., *Personal space*. 1969, Englewood Cliffs, NJ: Prentice-Hall.
- [26] Streitz, N.A., Geißler, J., Holmer, T., Konomi, S., Müller-Tomfelde, C., Reischl, W., Rexroth, P., P. Seitz, R., Steinmetz, *i-LAND: An interactive Landscape for Creativity and Innovation*, CHI'99, 120-127.
- [27] Swindells, C., Inkpen, K., Dill, J.C., Tory, M., *That one there! Pointing to establish device identity*, UIST 2002, 151-160.
- [28] Tandler, Prante, Müller-Tomfelde, Streitz, & Steinmetz, *Connectables: dynamic coupling of displays for the flexible creation of shared workspaces*, UIST 2001, 11-20.
- [29] Ullmer, B., Ishii, H., Glas, D., *mediaBlocks: Physical Containers, Transports, and Controls for Online Media*, SIGGRAPH'98, 379-386.